We are very grateful to the constructive comments. Now, we give responses for all questions/comments.

To reviewer #1: Thanks for your constructive comments to enhance the completeness of our work.

Q1: The comparison of validation error between search network and child network... The search progress...

R1: For different models, Table 1 in this response reports the comparison of the validation errors at the end of search and after architecture derivation without fine-tuning. The results show that DATA (gap=1.87, err=9.21) is superior to maintain better performance compared with SNAS (gap=2.13, err=9.33) and especially DARTS (gap=33.03, err=45.34), while requiring less training epochs than SNAS (100 vs 150 epochs), as the search progresses illustrated in Figure 2.

Q2: It is complained that NAS approach has high variance over different initializations... A variance plot...

R2: In Table 2 and Figure 3, the stds of DATA and the variances of DATA with different sampling time (M) are reported, which indicate that DATA (M=1,std=0.08) achieves lower stds than DARTS (k=1,std=0.14). Furthermore, Figure 4 shows the variances of architectures in search process for a comprehensive analysis. Please kindly check them.

Q3: Search result of M=4 and M=7.

R3: The search result (M=4) is shown in Figure 1. More results and analyses will be included in an updated version.

Q4: The proposed method is sampling based... And the gradient estimator should be written...

R4: Thanks for the suggestion, we will write the gradient estimator in an updated version. In brief, we have:

$$\frac{\partial f_{P_k^{(i,j)}}(G^{(i,j)})}{\partial P_k^{(i,j)}} = \frac{\frac{\partial}{\partial P_k^{(i,j)}} \exp\Big(\big(\log P_k^{(i,j)} + G_k^{(i,j)}\big)/\tau\Big)}{\sum_{l=1}^{K} \exp\Big(\big(\log P_l^{(i,j)} + G_l^{(i,j)}\big)/\tau\Big)} \left(\delta(k'-k) - \frac{\exp\Big(\big(\log P^{(i,j)} + G^{(i,j)}\big)/\tau\Big)}{\sum_{l=1}^{K} \exp\Big(\big(\log P_l^{(i,j)} + G_l^{(i,j)}\big)/\tau\Big)}\right)$$

$$= \frac{\delta(k'-k) - f_{P_k^{(i,j)}}(G^{(i,j)})}{\tau P_k^{(i,j)}} f(G_k^{(i,j)}),$$

where $G_k^{(i,j)}$ is the k-th Gumbel random variable, $P_k^{(i,j)}$ is the k-th element in $\mathbf{P}^{(i,j)} \in \mathbb{R}^K$ and it denotes the probability of choosing the k-th operation on the edge $e^{(i,j)}$. Then, the gradient of the loss with respect to other parameters can be easily calculated using the chain rule, since the other components in the network are differentiable as well.

Q5: Since the objective of architecture weights and probability vector are both training loss... is overfitting an issue...

R5: Similar to the dropout to combat overfitting, our EGS provides a network sampling technique that can be treated as a specific regularization, which is in agreement with the motivation in [C1]. That is, EGS endows DATA with the capability of searching architectures by learning on the sampled networks. Empirically, our experiments have strongly evidenced its validity. For a clearer verification, the example you mentioned will be included in an updated version.

Q6: DATA should be slower in terms per search iteration than SNAS... can authors provide more intuition on this part.

R6: The main reason is that only child architectures are sampled and optimized in DATA, which is faster than optimizing the whole network in SNAS. As shown in Figure 2, DATA requires less training epochs than SNAS (100 vs 150).

Please kindly check and reassess our paper based on these requested experimental results. Thanks.

Table 1: Validation error on CIFAR-10.

Model	Search	Child	Gap
DARTS	12.33	45.34	33.01
SNAS	11.46	9.33	2.13
DATA (M=7)	**11.08**	**9.21**	**1.87**

Table 2: Number of operations on CIFAR-10.

Model	Error (%)	Params (M)
DARTS(k=1)	3.00 ± 0.14	3.30
DARTS(k=2)	3.10 ± 0.12	4.00
DARTS(k=3)	2.95 ± 0.13	5.20
SNAS	2.85 ± 0.02	2.80
DATA(M=1)	2.95 ± 0.08	**2.59**
DATA(M=4)	2.70 ± 0.10	3.11
DATA(M=7)	$\mathbf{2.59} \pm \mathbf{0.09}$	3.74

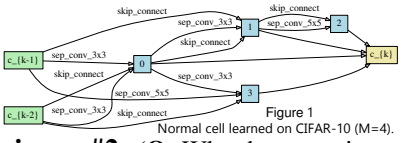
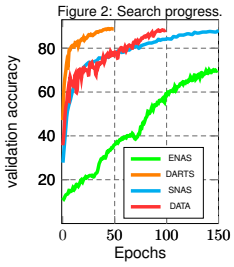
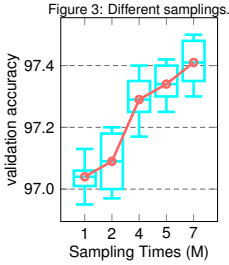
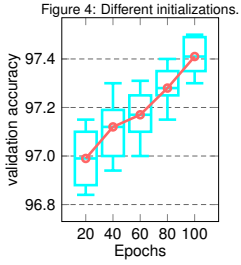


Figure 1
Normal cell learned on CIFAR-10 (M=4).







To reviewer #2: (**Q:** Why the operations used in each edge should be mutually exclusive)

R: As explained at line125 in the paper, DATA is actually inspired by the fact that operations may NOT be mutually exclusive but compatible, in contrast to SNAS and DARTS. In SNAS, any probability vector literally converges to an one-hot vector, which means only one operation is selected. Though DARTS is capable of selecting the top-k strongest operations, it always subjects to two limitations. First, the number of selected operations k is fixed on each edge, while DATA can adaptively adjust the numbers on different edges. Second, DATA (gap=1.87) can seamlessly bridge the gap between architectures in searching and validating, which obviously excels DARTS (gap=33.01), as shown in Table 1.

To reviewer #2 and #3: (**Q:** Comparison with a baseline that selects top-k operations during searching)

R: The results of selecting top-k operations using DARTS are reported in Table 2, which show that DARTS tends to select superfluous while inefficient operations that may dramatically introduce more parameters. In contrast, our EGS endows DATA with the capability of learning simple yet powerful operations, *e.g.*, the skip connection in Figure 1.

To reviewer #3: (**Q:** It would be interesting to know what other methods the authors tried... any convergence...)

R: Thanks for your interest. We have in the first place explored some naïve methods of sampling multiple operations on each edge, *i.e.*, selecting top-k operations. However, according to the issues as explained in our response to reviewer #2 and the results in Table 2, none of them works well in practice. For the convergence of DATA, we verify it empirically in Figure 2 and 4 in this response. It can be seen that our method converges favorably well.

[C1] Bender, Gabriel, *et al.* "Understanding and simplifying one-shot architecture search." ICML. 2018.